

Notice of Allowability

Application No.

09/913,003

Examiner

Zachary A Davis

Applicant(s)

MAO, WENBO

Art Unit

2137

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address--

All claims being allowable, PROSECUTION ON THE MERITS IS (OR REMAINS) CLOSED in this application. If not included herewith (or previously mailed), a Notice of Allowance (PTOL-85) or other appropriate communication will be mailed in due course. **THIS NOTICE OF ALLOWABILITY IS NOT A GRANT OF PATENT RIGHTS.** This application is subject to withdrawal from issue at the initiative of the Office or upon petition by the applicant. See 37 CFR 1.313 and MPEP 1308.

1. ☒ This communication is responsive to the second preliminary amendment filed 04 February 2002.
2. ☒ The allowed claim(s) is/are 1-9, 11-13, 15, and 16.
3. ☐ The drawings filed on _____ are accepted by the Examiner.
4. ☒ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
 - a) ☒ All b) ☐ Some* c) ☐ None of the:
 1. ☒ Certified copies of the priority documents have been received.
 2. ☐ Certified copies of the priority documents have been received in Application No. _____.
 3. ☐ Copies of the certified copies of the priority documents have been received in this national stage application from the International Bureau (PCT Rule 17.2(a)).

* Certified copies not received: _____.

Applicant has THREE MONTHS FROM THE "MAILING DATE" of this communication to file a reply complying with the requirements noted below. Failure to timely comply will result in ABANDONMENT of this application.

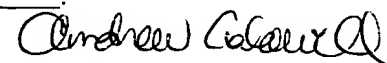
THIS THREE-MONTH PERIOD IS NOT EXTENDABLE.

5. ☐ A SUBSTITUTE OATH OR DECLARATION must be submitted. Note the attached EXAMINER'S AMENDMENT or NOTICE OF INFORMAL PATENT APPLICATION (PTO-152) which gives reason(s) why the oath or declaration is deficient.
6. ☒ CORRECTED DRAWINGS (as "replacement sheets") must be submitted.
 - (a) ☐ including changes required by the Notice of Draftsperson's Patent Drawing Review (PTO-948) attached
 - 1) ☐ hereto or 2) ☐ to Paper No./Mail Date _____.
 - (b) ☒ including changes required by the attached Examiner's Amendment / Comment or in the Office action of Paper No./Mail Date _____.

Identifying indicia such as the application number (see 37 CFR 1.84(c)) should be written on the drawings in the front (not the back) of each sheet. Replacement sheet(s) should be labeled as such in the header according to 37 CFR 1.121(d).
7. ☐ DEPOSIT OF and/or INFORMATION about the deposit of BIOLOGICAL MATERIAL must be submitted. Note the attached Examiner's comment regarding REQUIREMENT FOR THE DEPOSIT OF BIOLOGICAL MATERIAL.

Attachment(s)

1. ☒ Notice of References Cited (PTO-892)
2. ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3. ☒ Information Disclosure Statements (PTO-1449 or PTO/SB/08),
Paper No./Mail Date 20020108
4. ☐ Examiner's Comment Regarding Requirement for Deposit
of Biological Material
5. ☐ Notice of Informal Patent Application (PTO-152)
6. ☐ Interview Summary (PTO-413),
Paper No./Mail Date _____.
7. ☒ Examiner's Amendment/Comment
8. ☒ Examiner's Statement of Reasons for Allowance
9. ☐ Other _____.



ANDREW CALDWELL
SUPERVISORY PATENT EXAMINER

EXAMINER'S AMENDMENT

1. An examiner's amendment to the record appears below. Should the changes and/or additions be unacceptable to applicant, an amendment may be filed as provided by 37 CFR 1.312. To ensure consideration of such an amendment, it **MUST** be submitted no later than the payment of the issue fee.

Authorization for this examiner's amendment was given in a telephone interview with Robert Popa on 16 March 2005.

The application has been amended as follows:

CANCEL Claims 10 and 14.

REPLACE Claims 1-9, 11-13, 15, and 16 with the rewritten claims on the following sheets.

Amendments to the Claims

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (currently amended) A method of exchanging digital public-key verification data whereby a first party enables a second party to obtain probabilistic evidence that a given public-key number n is the product of exactly two odd primes p and q , not ~~known~~ known to the second party, whose bit lengths $(\ell(p), \ell(q))$ differ by not more than d bits; the method including the following steps, all operations being to mod P unless specified mod n , the method being halted should any check fail;
 - a) said first party provides to said second party a number P such that P is a prime number and $n|(P-1)$;
 - b) said second party provides to said first party a number g where $g = f^{(P-1)/n} \bmod P$, $f < P$;
 - c) said first party provides to said second party numbers A and B , where $A = g^p \bmod P$ and $B = g^q \bmod P$;
 - d) said second party checks that $A \neq B$, $A \neq 1$ and $B \neq 1$; whereupon the following steps are repeated up to k times;
 - e) said second party selects a random number $h \in Z_n^*$ such that $\left(\frac{h}{n}\right) = -1$ and provides the number h to the first party;
 - f) said first party checks that $\left(\frac{h}{n}\right) = -1$ and selects two random numbers u and v such that $\ell(u) = \ell((p-1)/2)$, $\ell(v) = \ell((q-1)/2)$ and provides to said second party the values $U = g^{2^u}$, $V = g^{2^v}$, $H_U = B^{(h^u \bmod n)}$, $H_V = A^{(h^v \bmod n)}$,
and $H_{UV} = h^u h^v \bmod n$;
 - g) said second party sends a request to the first party that the first party provides to the second party values r and s , which the second party randomly specifies should be either:
 - (1) $r = u$ and $s = v$; or

$$(2) r = u + (p - 1)/2, s = v + (q - 1)/2$$

h) said first party provides the requested values r and s to the second party,

i) if the second party requested $r = u$ and $s = v$, the second party determines whether:

$$(1) \ell(r) \leq \lfloor \ell(n)/2 \rfloor \pm d, \quad \ell(s) \leq \lfloor \ell(n)/2 \rfloor \pm d,$$

$$(2) g^{2r \pm 1} \equiv Ug, \quad g^{2s \pm 1} \equiv Vg,$$

$$(3) B^{(h^r \bmod n)} \equiv H_U, \quad A^{(h^s \bmod n)} \equiv H_V, \text{ and}$$

$$(4) h^r h^s \equiv H_{UV} \pmod{n};$$

thereby verifying the values provided by the first party to the second party are as were required by steps a) to f); or, if the second party requested

$r = u + (p - 1)/2, s = v + (q - 1)/2$, the second party determines whether:

$$(1) \ell(r) \leq \lfloor \ell(n)/2 \rfloor \pm d, \quad \ell(s) \leq \lfloor \ell(n)/2 \rfloor \pm d,$$

$$(2) g^{2r \pm 1} \equiv UA, \quad g^{2s \pm 1} \equiv VB,$$

$$(3) B^{(h^r \bmod n)} \equiv H_U^{\pm 1}, \quad A^{(h^s \bmod n)} \equiv H_V^{\mp 1} \quad (\pm \text{ and } \mp \text{ meaning the two}$$

exponents are of opposite sign), and

$$(4) h^r h^s \equiv H_{UV} h^{(n-1)/2} \pmod{n},$$

thereby obtaining said probabilistic evidence on whether the given public-key number n is the product of exactly two odd primes p and q whose bit lengths $(\ell(p), \ell(q))$ differ by not more than d bits.

2. (original) A method as claimed in claim 1 in which $d \leq 2$.

3. (currently amended) A method as claimed in claim 1 in which at least one of (1) the selections of random numbers is uniformly distributed, or (2) the choice of r and s is uniformly distributed.

4. (currently amended) A computing entity comprising:

a data processing equipment;

a memory; and

a communications equipment,

said data processing equipment being configured so as to be capable of processing data according to a set of instructions stored in said memory;

said communications equipment configured so as to communicate data according to said set of instructions such that the computing entity is configured to execute the following steps, wherein n is the product of exactly two odd primes p and q whose bit lengths $(\ell(p), \ell(q))$ differ by not more than d bits and all operations are to mod P unless specified mod n :

a) receive from another computing entity a number P such that P is a prime number and $n|(P-1)$;

b) provide to said other computing entity a number g where $g = f^{(P-1)/n} \bmod P$, $f < P$;

c) receive from said other computing entity numbers A and B , where $A = g^p \bmod P$ and $B = g^q \bmod P$;

d) check that $A \not\equiv B$, $A \not\equiv 1$ and $B \not\equiv 1$, and, if correct, repeat up to k times steps e) through i);

e) select a random number $h \in Z_n^*$ such that $\left(\frac{h}{n}\right) = -1$ and

provide the number h to said other computing entity;

f) receive from said other computing entity $U = g^{2u}$, $V = g^{2v}$, $H_U = B^{(h^u \bmod n)}$, $H_V = A^{(h^v \bmod n)}$, and $H_{UV} = h^u h^v \bmod n$ ~~entity were~~ where u and v are two random numbers such that

$\ell(u) = \ell((p-1)/2)$, $\ell(v) = \ell((q-1)/2)$;

g) request the other computing entity to provide values r and s , randomly specified to be either:

(1) $r = u$ and $s = v$; or

(2) $r = u + (p-1)/2$, $s = v + (q-1)/2$;

h) receive the requested values r and s from the other computing entity,

i) if $r = u$ and $s = v$ was requested, determine whether:

(1) $\ell(r) \leq \lfloor \ell(n)/2 \rfloor \pm d$, $\ell(s) \leq \lfloor \ell(n)/2 \rfloor \pm d$,

(2) $g^{2r+1} \equiv Ug$, $g^{2s+1} \equiv Vg$,

$$(3) \quad B^{(h^r \bmod n)} \equiv H_U, \quad A^{(h^s \bmod n)} \equiv H_V, \text{ and}$$

$$(4) \quad h^r h^s \equiv H_{UV} \pmod{n};$$

thereby verifying the values provided by the other computing entity are as were required by steps a) to i); or, if $r = u + (p-1)/2$, $s = v + (q-1)/2$ was requested, determine whether:

$$(1) \quad \ell(r) \leq \lfloor \ell(n)/2 \rfloor \pm d, \quad \ell(s) \leq \lfloor \ell(n)/2 \rfloor \pm d,$$

$$(2) \quad g^{2r+1} \equiv UA, \quad g^{2s+1} \equiv VB,$$

$$(3) \quad B^{(h^r \bmod n)} \equiv H_U^{\pm 1}, \quad A^{(h^s \bmod n)} \equiv H_V^{\mp 1} \quad (\pm \text{ and } \mp \text{ meaning the two exponents are of opposite sign), and}$$

$$(4) \quad h^r h^s \equiv H_{UV} h^{(n-1)/2} \pmod{n}$$

thereby obtaining said probabilistic evidence on whether the given public-key number n is the product of said exactly two odd primes p and q whose bit lengths ($\ell(p)$, $\ell(q)$) differ by not more than d bits.

5. (currently amended) A ~~method~~ computing entity as claimed in claim 4 in which $d \leq 2$.

6. (currently amended) A ~~method~~ computing entity as claimed in claim 4 in which at least one of (1) the selections of random numbers is uniformly distributed, or (2) the choice of r and s is uniformly distributed.

7. (currently amended) A computing entity comprising:
a data processing equipment;
a memory; and
a communications equipment,
said data processing equipment being configured so as to be capable of processing data according to a set of instructions stored in said memory;

said communications equipment configured so as to communicate data according to said set of instructions such that the computing entity is configured to execute the following steps, wherein n is the product of exactly two odd primes p and q whose bit lengths $(\ell(p), \ell(q))$ differ by not more than d bits and all operations are to mod P unless specified mod n :

- a) provide to another computing entity a number P such that P is a prime number and $n|(P-1)$;
- b) receive from the other computing entity a number g where $g = f^{(P-1)/n} \bmod P$, $f < P$;
- c) provide to said other computing entity numbers A and B , where $A = g^p \bmod P$ and $B = g^q \bmod P$;
- d) receive from said other computing entity a random number $h \in Z_n^*$ such that $\left(\frac{h}{n}\right) = -1$;
- e) check that $\left(\frac{h}{n}\right) = -1$ and, if so, select two random numbers u and v such that $\ell(u) = \ell((p-1)/2)$, $\ell(v) = \ell((q-1)/2)$ and provide to said other computing entity the values $U = g^{2u}$, $V = g^{2v}$, $H_U = B^{(h^u \bmod n)}$, $H_V = A^{(h^v \bmod n)}$ and $H_{UV} = h^u h^v \bmod n$;
- f) receive from said other computing entity a request to provide to said other computing entity values r and s , which said other computing entity randomly specifies should be either:
 - (1) $r = u$ and $s = v$; or
 - (2) $r = u + (p-1)/2$, $s = v + (q-1)/2$
- g) provide the requested values r and s to said other computing entity.

8. (currently amended) A ~~method~~ computing entity as claimed in claim 7 in which $d \leq 2$.

9. (currently amended) A ~~method~~ computing entity as claimed in claim 7 in which at least one of the selections of random numbers is uniformly distributed.

10. (canceled)

11. (currently amended) A communication system comprising at least a pair of computing entities as claimed in claim ~~10~~ 16 and a communications medium, each of said pair of computing

entities being arranged to communicate with the other computing entity via the communications medium.

12. (currently amended) A communication system as claimed in claim 11 in which said communications medium includes one or more of any of the internet, local area network, wide area network, virtual private circuit or public telecommunications network.

13. (currently amended) A computer storage medium having stored thereon a computer program readable by a general purpose computer, the computer program including instructions for said general purpose computer to configure it to be as said ~~computer~~ computing entity as claimed in claim 4.

14. (canceled)

15. (currently amended) A computer storage medium having stored thereon a computer program readable by a general purpose computer, the computer program including instructions for said general purpose computer to configure it to be as said ~~computer~~ computing entity as claimed in claim 7.

16. (currently amended) A system of co-operating computer entities, including:
a first computing entity comprising:[:]
a data processing equipment;
a memory; and
a communications equipment;
said data processing equipment being configured so as to be capable of processing data according to a set of instructions stored in said memory;
said communications equipment configured so as to communicate data according to said set of instructions such that the first computing entity is configured to execute the following steps, wherein n is the product of exactly two odd primes p and q whose bit lengths $(\ell(p), \ell(q))$ differ by not more than d bits and all operations are to mod P unless specified mod n :

a) receive from ~~another~~ a second computing entity a number P such that P is a prime number and $n|(P-1)$;

b) provide to said ~~other~~ second computing entity a number g where $g = f^{(P-1)/n} \bmod P$, $f < P$;

c) receive from said ~~other~~ second computing entity numbers A and B , where $A = g^p \bmod P$ and $B = g^q \bmod P$;

d) check that $A \neq B$, $A \neq 1$ and $B \neq 1$, and, if correct, repeat up to k times steps e) through i);

e) select a random number $h \in Z_n^*$ such that $\left(\frac{h}{n}\right) = -1$ and provide the number h to said ~~other~~ second computing entity;

f) receive from said ~~other~~ second computing entity $U = g^{2u}$, $V = g^{2v}$, $H_U = B^{\left(h^u \bmod n\right)}$,

$H_V = A^{\left(h^v \bmod n\right)}$, and $H_{UV} = h^u h^v \bmod n$ ~~entity were~~ where u and v are two random numbers such that $\ell(u) = \ell((p-1)/2)$, $\ell(v) = \ell((q-1)/2)$;

g) request the ~~other~~ second computing entity to provide values r and s , randomly specified to be either:

(1) $r = u$ and $s = v$; or

(2) $r = u + (p-1)/2$, $s = v + (q-1)/2$;

h) receive the requested values r and s from the ~~other~~ second computing entity,

i) if $r = u$ and $s = v$ was requested, determine whether:

(1) $\ell(r) \leq \left\lfloor \ell(n)/2 \right\rfloor \pm d$, $\ell(s) \leq \left\lfloor \ell(n)/2 \right\rfloor \pm d$,

(2) $g^{2r+1} \equiv Ug$, $g^{2s+1} \equiv Vg$,

(3) $B^{\left(h^r \bmod n\right)} \equiv H_U$, $A^{\left(h^s \bmod n\right)} \equiv H_V$, and

(4) $h^r h^s \equiv H_{UV} \pmod{n}$;

thereby verifying the values provided by the ~~other~~ second computing entity are as were required by steps a) to i); or, if $r = u + (p-1)/2$, $s = v + (q-1)/2$ was requested, determine whether:

(1) $\ell(r) \leq \left\lfloor \ell(n)/2 \right\rfloor \pm d$, $\ell(s) \leq \left\lfloor \ell(n)/2 \right\rfloor \pm d$,

(2) $g^{2r+1} \equiv UA$, $g^{2s+1} \equiv VB$,

(3) $B^{(h^r \bmod n)} \equiv H_U^{\pm 1}$, $A^{(h^s \bmod n)} \equiv H_V^{\mp 1}$ (\pm and \mp meaning the two exponents are of opposite sign), and

$$(4) h^r h^s \equiv H_{UV} h^{(n-1)/2} \pmod{n}$$

thereby obtaining said probabilistic evidence on whether the given public-key number n is the product of said exactly two odd primes p and q whose bit lengths $(\ell(p), \ell(q))$ differ by not more than d bits; and

[[a]] said second computing entity comprising:

a data processing equipment;

a memory; and

a communications equipment;

said data processing equipment being configured so as to be capable of processing data according to a set of instructions stored in said memory;

said communications equipment configured so as to communicate data according to said set of instructions such that the second computing entity is configured to execute the following steps, wherein n is the product of exactly two odd primes p and q whose bit lengths $(\ell(p), \ell(q))$ differ by not more than d bits and all operations are to mod P unless specified mod n :

a) provide to another said first computing entity [[a]] the number P ;

b) receive from the other first computing entity [[a]] the number g where $g = f^{c(P-1)/n} \bmod P$, $f < P$;

c) provide to said other first computing entity numbers A and B , where $A = g^p \bmod P$ and $B = g^q \bmod P$;

d) receive from said other first computing entity [[a]] the random number $h \in Z_n^*$ such that $\left(\frac{h}{n}\right) = -1$;

e) check that $\left(\frac{h}{n}\right) = -1$ and, if so, select two random numbers u and v such that

$\ell(u) = \ell((p-1)/2)$, $\ell(v) = \ell((q-1)/2)$ and provide to said other first computing entity the

values of $U = g^{2u}$, $V = g^{2v}$, $H_U = B^{(h^u \bmod n)}$, $H_V = A^{(h^v \bmod n)}$, and $H_{UV} = h^u h^v \bmod n$

f) receive from said ~~other~~ first computing entity a request to provide to said ~~other~~ first computing entity values r and s , which said first computing entity randomly specified should be either:

(1) $r = u$ and $s = v$; or

(2) $r = u + (p - 1)/2$, $s = v + (q - 1)/2$; and

g) provide the requested values r and s to said ~~other~~ first computing entity.

Application/Control Number: 09/913,003

Art Unit: 2137

Page ¹²~~7~~

REPLACE pages 3, 11, 12, 14, 15, 19, and 28 of the specification with the rewritten specification pages on the following sheets.

Thus, k should be in thousands ($k = 3000$ was suggested in [R. Berger, S. Kannan and R. Peralta. A framework for the study of cryptographic protocols, *Advances in Cryptology – Proceedings of CRYPTO 85* (H.C. Williams ed.), Lecture Notes in Computer Science, Springer-Verlag 218 (1986), pp. 87-103]) in order for the error probability to be negligibly small. (We note $e^{-3000/74} < 1/2^{58} < e^{-3000/75}$ and regard an amount at this level to be negligibly small). Since the cost for computing a square root mod n is measured by $O(\log_2 n)$ multiplications of integers mod n , the total cost for proving the two-prime-product structure of a number n by showing quadratic residue information will be $O(k \log_2 n)$ (multiplications mod n) with an error probability between $e^{-k/74}$ and $e^{-k/75}$.

Van de Graaf and Peralta [J. van de Graaf and R. Peralta. A simple and secure way to show the validity of your public-key, *Advances in Cryptology – Proceedings of CRYPTO 87* (E. Pomerance, ed.), Lecture Notes in Computer Science, Springer-Verlag 293 (1988), pp. 128-134] observed that if n is a Blum integer, that is, n is the product of two distinct prime factors (again this may include their powers), both congruent to 3 mod 4, then any element in the multiplicative group mod n with the positive Jacobi symbol has the property that either itself or its negation is a quadratic residue modulo n . Their protocol for proof of Blum integer is based on this fact. A number of other previous protocols for proving two-prime-product structure also use this idea (e.g., [J. Camenisch and M. Michels. Proving in zero-knowledge that a number is the product of two safe primes, In *Advances in Cryptology – EUROCRYPT 99*, Lecture Notes in Computer Science, Springer-Verlag 1592 (1999), pp. 106-121, R. Gennaro, D. Miccianicio and T. Rabin. An efficient non-interactive statistical zero-knowledge proof system for quasi-safe prime products, In 5th ACM Conference on Computer and Communications Security, October 1998, M. Liskov and R.D. Silverman. A statistical limited-knowledge proof for secure RSA keys, IEEE P1363 Research Contributions]). Note that provided n is not a square number (which is easy to test against), exactly half of the elements in the multiplicative group mod n can have a positive Jacobi symbol which is also easy to evaluate. Thus, given such n , the above demonstration actually shows that a quarter of elements in the group are quadratic residues (since a quadratic residue must have positive Legendre symbol mod all prime factors, and only half of elements mod a prime

Referring to Fig 1, there is illustrated schematically a pair of computing entities 102,104 configured for communicating electronic data with each other over a communications network, in this case the internet 106, by communicating data 108,110 to each other via the internet 106 in well known manner. Illustrated in Fig 1 is first computing entity 102, herein after referred to as entity A and a second computing entity 104 herein referred to as entity B. In the example illustrated in Fig 1, the first and second computing entities 102,104 are geographically remote from each other and whilst in the best mode herein, the communications network comprises the known internet 106, in other embodiments and implementations of the present invention the communications network could comprise any suitable means of transmitting digitized data between the computing entities. For example, a known Ethernet network, local area network, wide area network, virtual private circuit or public telecommunications network may form the basis of a communications medium between the computing entities 102,104.

The computing entities 102 and 104 have been programmed by storing on memory programs read from computer program storage media 112,114, for example, a CD-ROM.

Referring now to Fig 2, there is illustrated schematically physical resources and logical resources of the computing entities A and B. Each computing entity comprises at least one data processing means 200,202, a memory area 203,205, a communications port 206,208, for communicating with other computing entities. There is an operating system 209,211, for example a known Unix operating system. One or more applications programs 212, 214 are configured for operating for receiving, transmitting and performing data processing on electronic data received from other computing entities, and transmitted to other computer entities in accordance with specific methods of the present invention. Optionally there is a user interface 215,217 which may comprise a visual display device, a pointing device, e.g. a mouse or track-ball device, a keypad, and a printer.

Under control of the respective application program 212,214, each of the computing entities 102, 104 is configured to operate according to a first specific method of the present invention.

Referring to Fig 3 herein, there is illustrated schematically data communications passed between the first and second computing entities to effect verification by B of A's

private components of a public-key according to the first specific implementation of the present invention.

Applications programs 212,214 operate a set of algorithms that effect implementation of the verification protocol. The precise implementation of the algorithms is preferably made in a conventional prior art programming language, for example the language C, or C++ using conventional programming techniques which are known to those skilled in the art. For a better understanding of the implementation of the algorithms, the following presents a model, notation and explanation of the verification protocol. It will be understood by those skilled in the art that the algorithmic steps are used to control the logical and physical resources of the computing entities by being programmed into the applications in a conventional programming language.

Referring now to Figures 3, 4 and 5, there will now be described the operation of two computing entities commonly referred to Alice and Bob which will be adopted here. The computing entity 102 and computing entity 104 by following the steps of Figures 4 and 5, respectively, exchanging signals representative of various data values as shown in Figure

First, the computing entities agree on a set of parameters as follows, where Alice 102 is the prover and Bob 104 is the verifier. Alice has constructed $n = pq$ such that p and q are distinct odd primes with $|\ell(p) - \ell(q)| \leq d$ (i.e., the lengths of the two primes differ by at most d bits). The length of n is generally at least 512 to meet common security standards. d is preferably no greater than 2 but can be larger. The disadvantage of a larger d is that as d increases it will reach a threshold where the probability of p and q are primes when the test of the present invention is passed becomes dependent on n not k .

A proof will be abandoned on Alice's instigation if any check she (i.e., the computing entity A) performs fails and will be rejected by Bob if any check he (ie the computing entity B) performs fails.

First, Alice shall help Bob to set up a multiplicative group of order n . For her part, Alice only needs to generate a prime P with $n|(P - 1)$. This prime can be constructed by testing the primality of $P = 2\alpha n + 1$ for $\alpha = 1, 2, \dots$, until P is found to be prime. By the prime number theorem (general form due to Dirichlet, see e.g., p.28 of [E. Kranakis.

For clarity, we shall omit the trailing mod P operation in the following protocol specification which, for reference will be called Two_Prime_Product (n, g, A, B, P)

The following steps are repeated k times

1. Bob picks $h \in Z_n^*$ at random with $\left(\frac{h}{n}\right) = -1$ (step 520) and sends it to Alice (steps 522, data 308).
2. Alice receives h (step 420) and checks $\left(\frac{h}{n}\right) = -1$ (step 422) and abandons the proof (step 424) if the check fails. If the check passes Alice, picks u, v at random (step 426) such that

$$\ell(u) = \ell((p-1)/2)$$

$$\ell(v) = \ell((q-1)/2)$$

and sets

$$U = g^{2u}, \quad V = g^{2v}, \quad H_U = B^{(h^u \bmod n)},$$

$$H_V = A^{(h^v \bmod n)}, \quad H_{UV} = h^u h^v \bmod n \quad (\text{step 427});$$

Alice sends to Bob: U, V, H_u, H_v, H_{uv} (steps 428, data 310).

3. Bob receives these values from A (step 524) and picks a challenge $c \in \{0,1\}$ at random (step 526) and sends it to Alice (steps 528, data 312).
4. Alice receives the challenge (step 430) and sends Bob the responses

$$r = u + c(p-1)/2, \quad s = v + c(q-1)/2 \quad (\text{step 432, data 314}).$$

5. Bob receives r and s from A (step 530) checks all of the following ($c = 0$);

$$5.1 \quad \ell(r) \leq \lfloor \ell(n)/2 \rfloor + 2, \quad \ell(s) \leq \lfloor \ell(n)/2 \rfloor + 2$$

$$5.2 \quad g^{2r+1} \equiv Ug, \quad g^{2s+1} \equiv Vg$$

$$5.3 \quad B^{(h^r \bmod n)} \equiv H_U, \quad A^{(h^s \bmod n)} \equiv H_V$$

$$5.4 \quad h^r h^s \equiv H_{UV} \pmod{n}$$

6. Bob checks: ($c = 1$)

$$6.1 \quad \ell(r) \leq \lfloor \ell(n)/2 \rfloor + 2, \quad \ell(s) \leq \lfloor \ell(n)/2 \rfloor + 2$$

$$6.2 \quad g^{2r+1} \equiv UA, \quad g^{2s+1} \equiv VB$$

$$6.3 \quad B^{(h^r \bmod n)} \equiv H_U^{\pm 1}, \quad A^{(h^s \bmod n)} \equiv H_V^{\mp 1}$$

($H_U^{\pm 1}$ and $H_V^{\mp 1}$ means the exponents take opposite signs)

$$6.4 \quad h^r h^s \equiv H_{UV} h^{(n-1)/2} \pmod{n}$$

at steps 534 or 536, for $c = 0, c = 1$, respectively.

If $c = 0$ and any of the checks of step 534 fails then Bob rejects the proof (step 536). Similarly, if $c = 1$ and if the checks of step 538 fail, then Bob rejects the proof (step 536).

If the checks at step 534 all pass, Bob decides if a primality check for a further value of n is required (step 540). If “Yes” Bob chooses another h (step 520) and another iteration is carried out; if “No” the protocol is ended (step 542).

If the checks at step 538 all pass, Bob checks for Monte-Carlo evidence at step 539 and then determines if another iteration is to be carried out (step 540).

Lemma 1 Without the knowledge of the factorization of n , the element g fixed by Bob satisfies

$$\Pr[\text{Ord}_P(g) \text{ divides } x] = x/n,$$

for any x divides n .

Proof Without the knowledge of the factorization of n , Bob's procedure for fixing g is via $g = f^{(P-1)/n} \bmod P$ using f which is chosen at random from Z_P^*

Then $g^n \equiv 1 \pmod{P}$ by Fermat's Theorem. In the cyclic group Z_P^* there are exactly $n = \sum_{d|n} \phi(d)$ elements of orders dividing n . Only these elements can be the candidates for g . For the same reason, for any $x | n | P-1$, there are exactly $x = \sum_{d|x} \phi(d)$ elements in Z_P^* of orders dividing x . The claimed probability is thus calculated as that of picking x objects from n . \square

Lemma 2 Denote $\text{Ord}_P(B) = x$ and $\text{Ord}_P(A) = y$. Upon acceptance of a proof on running `Two_Prime_Product`(n, g, A, B, P), Bob accepts that his random choice of h in the protocol run $((h, n) = 1 \text{ and } (\frac{h}{n}) = -1)$ satisfies

$$\begin{cases} h^{[(\log_g(A)-1)/2]} \equiv \pm 1 \pmod{x} \\ h^{[(\log_g(B)-1)/2]} \equiv \mp 1 \pmod{y} \end{cases},$$

and the probability for failing this does not exceed $1/2^k$ where k is the number of iterations used in the protocol.

Proof The first congruence in 5.2 shows that Alice knows both $\log_g(U)$ (shown when $c = 0$) and $\log_g(UA) = \log_g(U) + \log_g(A)$ (shown when $c = 1$), and has added $\log_g(A)$ to the response whenever $c = 1$ is the case. Suppose Alice does not know $\log_g(A)$. Then in each iteration she can only answer Bob's random challenge with at most $1/2$ chance of correctness. Thus, after having verified k times of correct responses to his random challenges, Bob should agree that the probability for Alice not having used $\log_g(A)$ in her response (when $c = 1$) is at most $1/2^k$.

The first congruence in 5.3 further shows that H_U is generated from B with the use of an exponent which is in turn generated from Bob's randomly chosen challenge h . Since $(h, n) = 1$, $(h^r \bmod n, n) = 1$. Therefore

$$\text{Ord}_P(H_U) = \text{Ord}_P(B) = x.$$

Clearly, the quantity $\log_g(A)$ in $2r + 1$ (when $c = 1$) amounts to $(\log_g(A) - 1)/2$ in r . Therefore the first congruence in 5.3 shows that for h satisfying $(h, n) = 1$:

$$h^{[(\log_g(A)-1)/2]} \equiv \pm 1 \pmod{x}.$$

Analogously we can use the second congruence in 5.3 to establish that for the same h

$$h^{[(\log_g(B)-1)/2]} \equiv \mp 1 \pmod{y}. \quad \square$$

In the rest of this section we will continue denoting

$$\text{Ord}_P(B) = x, \quad \text{Ord}_P(A) = y.$$

Since in bit operation, the cost for exponentiation mod P is measured in $O((\log_2 P)^3)$, i.e., $C(\log_2 P)^3$ for some constant C , we can use the following to relate the cost of exponentiation mod P to that mod n (of any size larger than 512 bits):

$$(\log_2 P)^3 \leq (1.02 \log_2 n)^3 \approx 1.062 (\log_2 n)^3$$

That is, the cost of one exponentiation mod P will not exceed that of one mod n by seven percent. We nevertheless use a ten percent expansion and convert Bob's workload of $5k$ exponentiations mod P into $5.5k$ exponentiations mod n . So in total Bob will need to compute no more than $7k$ of them. Since on average an exponentiation mod n amounts to $1.5 \log_2 n$ multiplications mod n , the total cost to Bob for running the protocol will be $12k \log_2 n$ multiplications of integer of size n . We can also use this quantity to bound Alice's cost of running the protocol.

For n of size larger than 512 bits, the computational cost of proving and verifying that n is the product of two primes of roughly equal size using protocol Two_Prime_Product is $12k \log_2 n$ multiplications of integer of size of n . Both parties should perform this number of operations.

Considering the fact that a Monte-Carlo primality test on non-secret number mainly involves modulo exponentiation, Bob's verification cost is equal to eight such tests on non-secret numbers of size n .

We have constructed an efficient knowledge proof protocol for demonstrating an integer being the product of two prime factors of roughly equal size. The new protocol is the first of its kind that proves such a structure with efficiency comparable to that of a Monte-Carlo method for primality evidence "in the dark".

Previous techniques for proving such a structure have a much higher cost for non-Blum integers (as will be discussed below). The improved efficiency for reasoning about non-Blum integers due to this work manifests a particular suitability for using the proposed protocol in the proof of valid RSA keys which are generated at uniformly random (e.g., for the protocol of Blackburn and Galbraith (S.R. Blackburn and S.D. Galbraith. Certification of secure RSA keys, University of Waterloo Centre for Applied Cryptographic Research, Technical Report CORR 90-44.

2. The following changes to the drawings have been approved by the examiner and agreed upon by applicant: the **PROPOSED DRAWING CORRECTIONS** to Figures 3, 4B, 5A, and 5C are submitted on the following sheets. In order to avoid abandonment of the application, applicant must make these above agreed upon drawing changes.

Annotated sheet
3/9

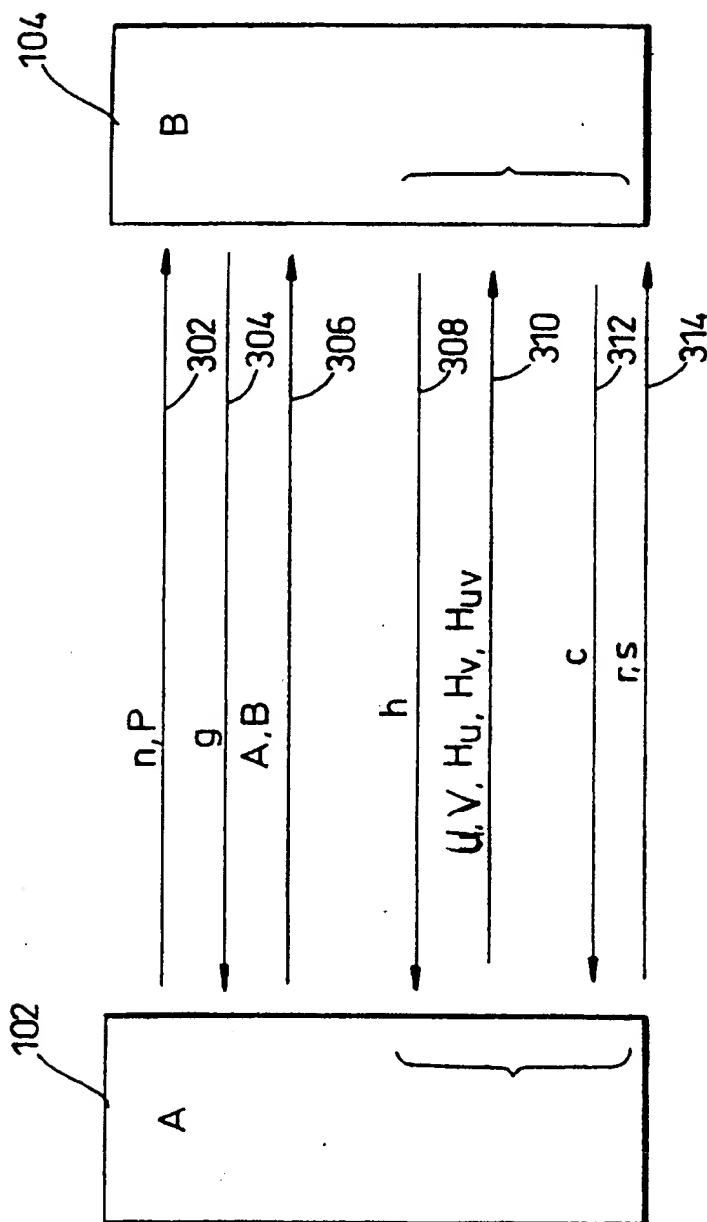
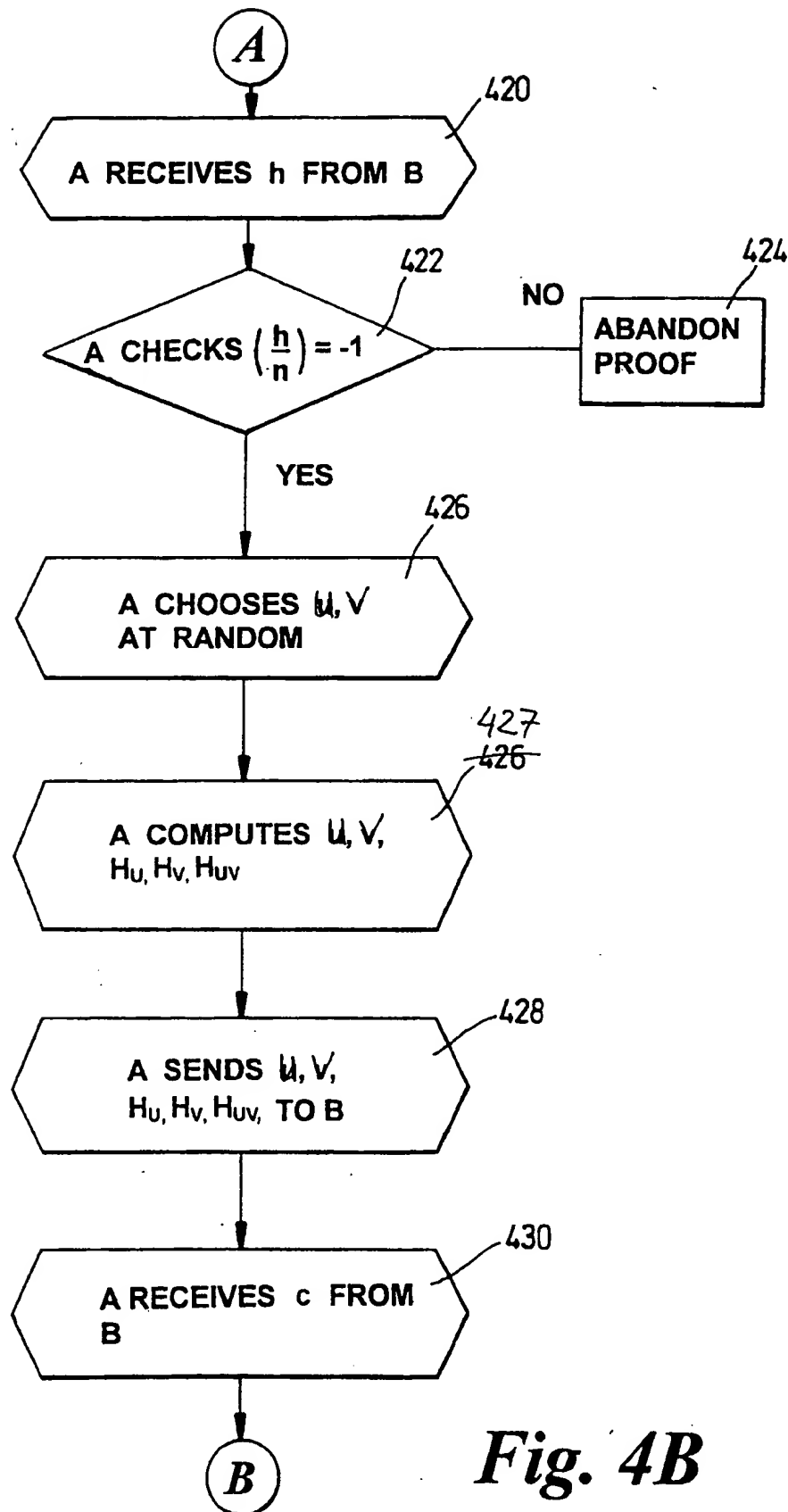


Fig. 3

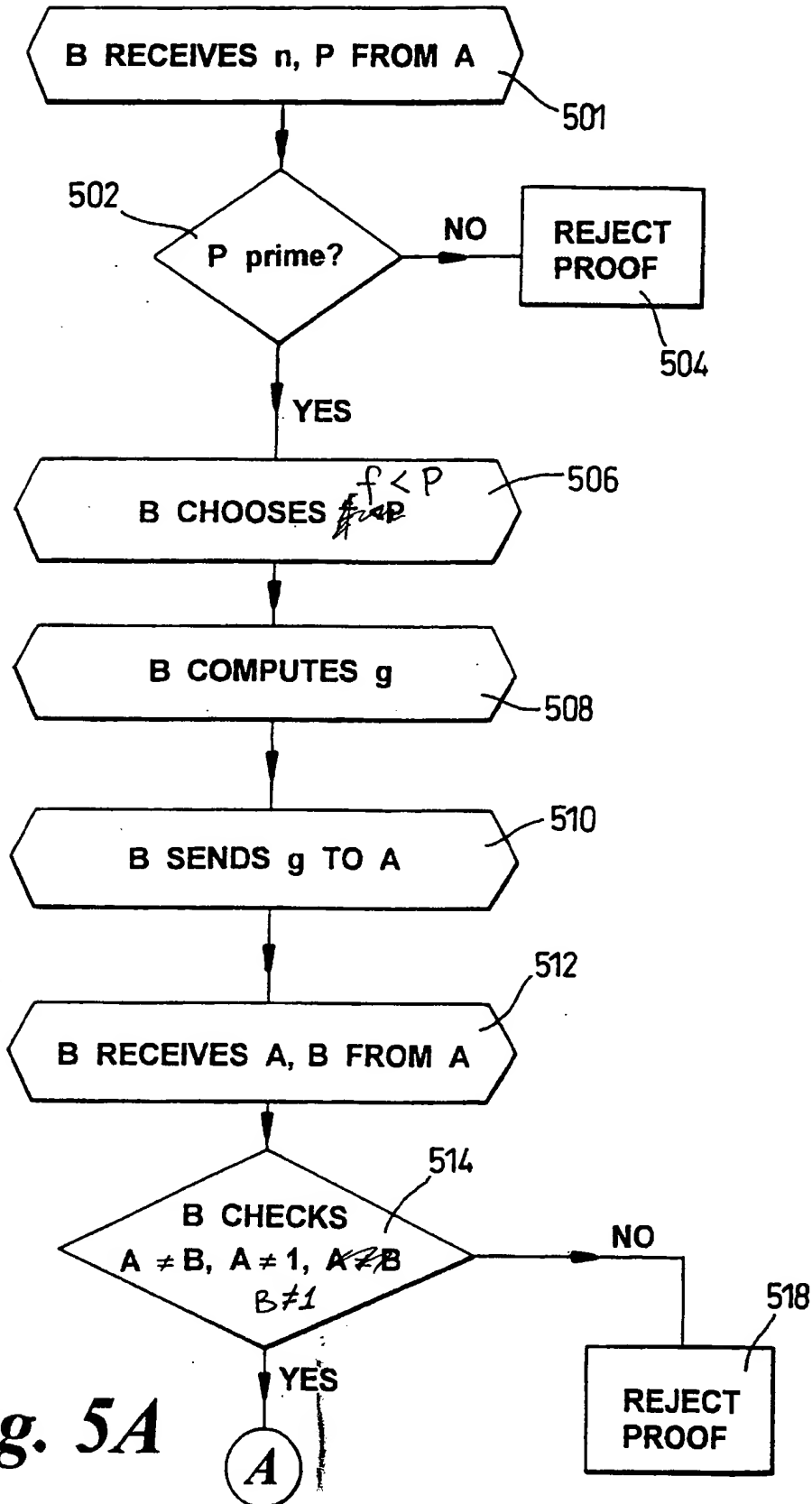
5/9

Annotated sheet

**Fig. 4B**

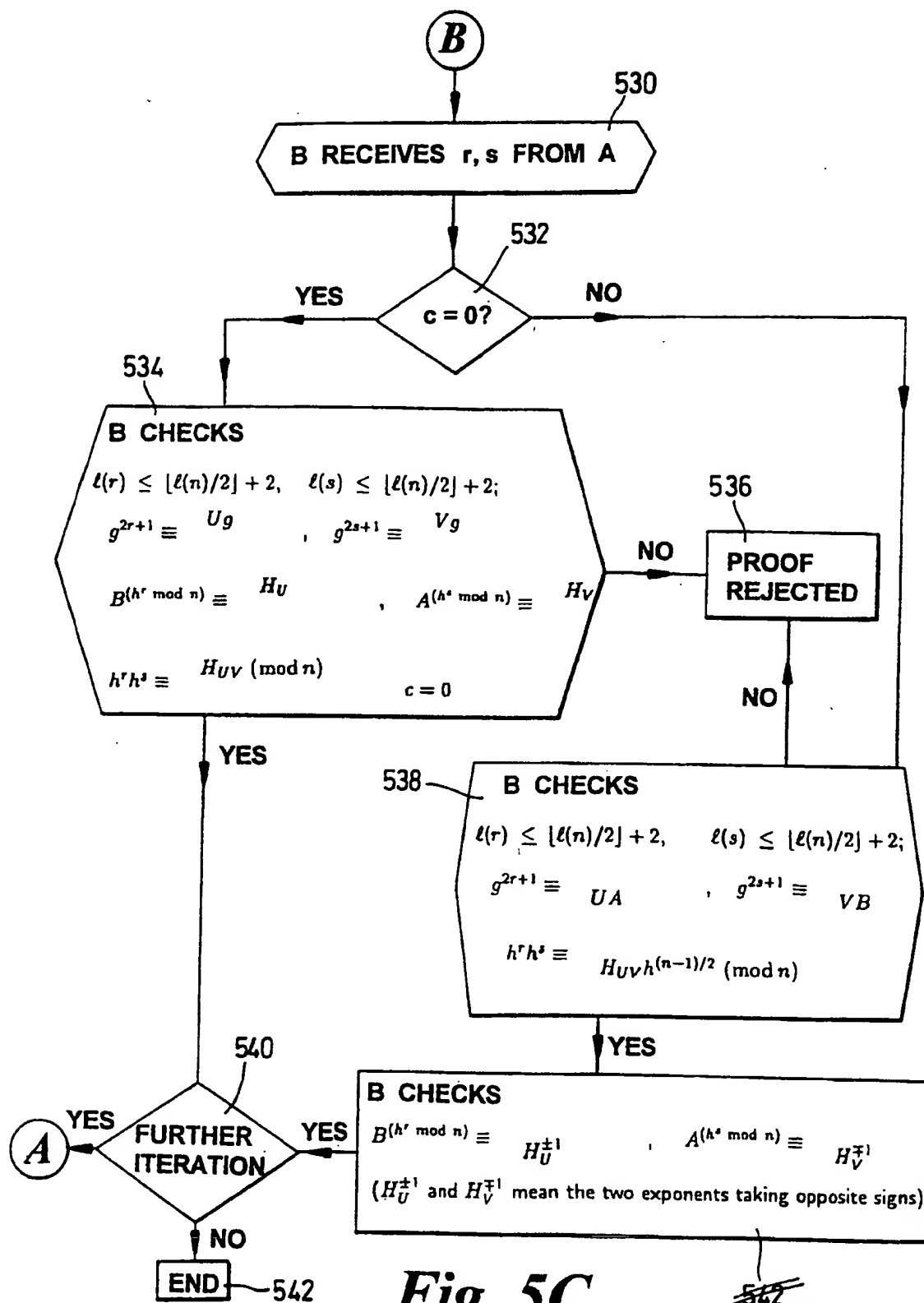
Annotated sheet

7/9

**Fig. 5A**

9/9

Annotated sheet



3. The following is an examiner's statement of reasons for allowance:

Independent Claim 1 is directed to a method for verifying portions of a public key using a zero-knowledge proof. The closest prior art, van de Graaf et al, "A Simple and Secure Way to Show the Validity of Your Public Key" (cited in Applicant's specification and included on the Information Disclosure Statement received 08 January 2002), also discloses a method for verifying a public key by zero-knowledge proof. However, the method of Claim 1 is distinguished from the method of van de Graaf in that the method of Claim 1 uses a specific Monte Carlo method for testing primality that is not disclosed by van de Graaf. Claim 1 is further distinguished because the method does not require the numbers being tested to be Blum integers, which is a condition required by the method of van de Graaf.

Independent Claims 4 and 7 are directed to computers implementing the method of Claim 1, and independent Claim 16 is directed to a system including the computers as claimed in Claims 4 and 7. These claims are therefore allowable for the reasons discussed above.

The Examiner notes that the document cited on the included form PTO-892, Mao, "Fast Monte-Carlo Primality Evidence Shown in the Dark", discloses the method as claimed. It does not qualify as prior art, as the author is the named Applicant, and the document was published after the earliest priority date. It is being included for the completeness of the prosecution history.

Any comments considered necessary by applicant must be submitted no later than the payment of the issue fee and, to avoid processing delays, should preferably

accompany the issue fee. Such submissions should be clearly labeled "Comments on Statement of Reasons for Allowance."

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Zachary A Davis whose telephone number is (571) 272-3870. The examiner can normally be reached on weekdays 8:30-6:00, alternate Fridays off.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Andrew Caldwell can be reached on (571) 272-3868. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

ZAD
zad



ANDREW CALDWELL
SUPERVISORY PATENT EXAMINER